

PROJEKTOWANIE ZINTEGROWANE NA POZIOMIE PROCESOROWYM

Andrzej Stasiak

**Instytut Informatyki i Elektroniki, Uniwersytet Zielonogórski
65-246 Zielona Góra, ul. Podgórna 50**

e-mail: A.Stasiak@iie.uz.zgora.pl

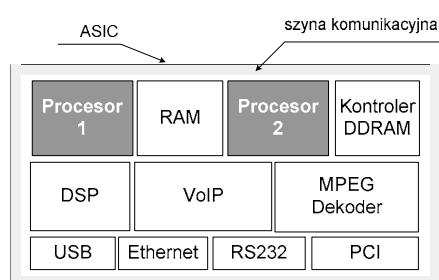
STRESZCZENIE

Proces dekompozycji systemowej rozumiany jako podział specyfikacji wejściowej systemu na część programową i sprzętową, jest jednym z nowych i ciągle rozwijanych zagadnień dotyczących projektowania systemów zintegrowanych. W zintegrowanym projektowaniu sprzętu i oprogramowania wyróżniamy dekompozycję systemową i procesorową. Artykuł omawia zagadnienia dotyczące projektowania sprzętowo-programowego mikrosystemu cyfrowego w zakresie dekompozycji procesorowej.

1. SYSTEMY CYFROWE

Na szerokim polu zastosowań inżyniersko-przemysłowych bezustannie występuje zapotrzebowanie na coraz bardziej wydajne systemy obliczeniowe. Opracowywane są wciąż nowe metody projektowania złożonych systemów cyfrowych w celu uproszczenia procesu poznawczego, korekty błędów oraz zarządzania. Dzisiejsze systemy cyfrowe składają się z wielu komponentów sprzętowych i programowych, które pozostają w ścisłej interakcji ze sobą. Określenie właściwego podziału (balansu) systemu pomiędzy obiema częściami determinuje sukces rozwiązania. Rozważając rozwiązanie programowe można łatwo zaobserwować, że komponenty programowe są prostsze do modyfikacji, niż ich odpowiedniki sprzętowe. Dzięki tej właściwości, programowe elementy systemu przetwarzane przez programowalne procesory, zapewniają prostotę i szybkość podczas eliminacji błędów, łatwą zmianę zastosowania danego programu, ponowne użycie funkcji/procedur do budowy innych programów. Jednak w porównaniu z komponentami sprzętowymi, elementy programowe są dużo wolniejsze i zużywają znacznie więcej energii podczas wykonywania tych samych funkcji. Rozwiązania wyłącznie sprzętowe używane są w systemach, gdzie czas reakcji i pobór mocy są parametrami krytycznymi. Niestety, czas projektowania takich systemów jest długi i drogi. Dodatkowo, procedury zaimplementowane w strukturach sprzętowych nie mogą być modyfikowane.

W procesie rozwoju zintegrowanych układów cyfrowych opracowano architekturę typu SoC (ang. *System on a Chip*). W tym rozwiązaniu wszystkie elementy składowe projektowanego systemu (CPU, pamięć, interfejs wej./wyj., inne) zostają zintegrowane w jednym układzie scalonym [1] typu ASIC (ang. *Application Specific Integrated Circuit*) (rysunek 1). Zadaniem projektanta systemu jest znalezienie właściwego balansu pomiędzy komponentami programowymi i sprzętowymi w celu zapewnienia wymagań pracy systemu i kosztów realizacji.



Rys. 1. Przykład systemu SoC

Główną zaletą układów SoC jest przede wszystkim redukcja kosztów realizacji projektu. Natomiast do cech charakterystycznych systemów SoC zaliczamy:

- małe gabaryty fizyczne; dzięki czemu możliwe jest projektowanie urządzeń przenośnych takich jak telefony komórkowe, PDA, palmtop, itp.,
- większa wydajność pracy poprzez zintegrowanie składowych systemu,
- zredukowany pobór mocy,
- krótki czas realizacji projektu.

Rozwiązaniem znacznie uniwersalnym oraz coraz częściej stosowanym jest architektura SOPC (ang. *System On a Programmable Chip*) [4]. Cechą charakterystyczną SOPC jest zintegrowanie w jednym układzie scalonym logiki reprogramowalnej FPGA z mikroprocesorem bądź mikrokontrolerem pełniącym rolę głównej jednostki sterowania (często również przetwarzania) w systemie. Blok CPU może zostać zaimplementowany jako element ASIC o rdzeniu stałym (tzw. ang. *hard core*) lub o rdzeniu zmiennym (tzw. ang. *soft core*) tj. jako komponent IP CORE (ang. *Intellectual Property CORE*). Główną cechą wyróżniającą układy SOPC jest możliwość swobodnego dokonywania modyfikacji struktury wewnętrznej układu cyfrowego nawet podczas pracy systemu.

Ponadto, budowa systemu z wykorzystaniem architektury SOPC umożliwia przeprowadzenie procesu aktualizacji funkcjonalności (oprogramowania definiującego zadania) części programowej i sprzętowej oraz naprawę błędów po procesie implementacji (produkcji) i wdrożeniu systemu. Są to systemy, które wykorzystuje się do budowy kompleksowych, pod względem funkcjonalności, systemów cyfrowych. Znajdują zastosowanie w wielu dziedzinach, między innymi w takich jak:

- telekomunikacja,
- przetwarzanie dźwięku i obrazu,
- systemy sterowania,
- specjalizowane systemy obliczeniowe.

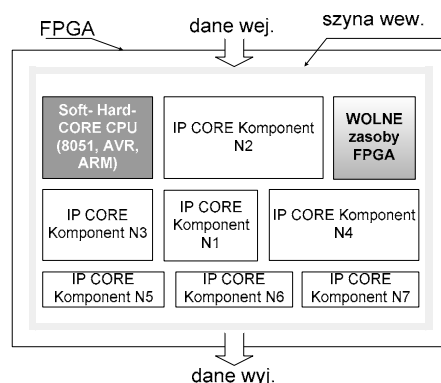
Proces projektowania systemów zintegrowanych (SoC, SOPC, inne architektury systemów osadzonych) nazywamy zintegrowanym projektowaniem sprzętu i oprogramowania (ang. *hardware-software codesign*) [2,3,6]. Jest to jeden z podstawowych etapów nowoczesnego podejścia do projektowania systemów cyfrowych [5]. W procesie projektowym system poddawany jest podziałowi na program i sprzęt dopiero w końcowej fazie procesu projektowego [3], a nie w fazie początkowej jak w podejściu tradycyjnym. W ten sposób możliwe jest podejmowanie kluczowych decyzji dotyczących podziału systemu na sprzęt i program w końcowym etapie projektowym, tj. w czasie, gdy znane są dokładne informacje dotyczące możliwości i ograniczeń pracy systemu dla zdefiniowanej architektury sprzętowej.

2. SPRZĘTOWO-PROGRAMOWY MIKROSYSTEM CYFROWY

W systemach zintegrowanych, cechą charakterystyczną głównej jednostki sterowania i przetwarzania - CPU, wynikającą z jej budowy, jest to, że wszystkie instrukcje (program) wykonywane są w sposób sekwencyjny. Przy dużych obciążeniach systemu (systemy reaktywne, systemy bezprzewodowe), sumaryczna wydajność całego systemu może spaść do poziomu krytycznego – poniżej minimum. Oczywiście, jest to efekt niepożądany. Poszukuje się rozwiązań w celu eliminacji powyższego problemu [6]. Rozwiązaniem jest dedykowany sprzętowy akcelerator dla układu CPU. Wśród propozycji poprawy efektywności systemów obliczeniowych przy użyciu dodatkowego komponentu sprzętowego interesujące są dwa sposoby realizacji zintegrowanego koprocatora/akceleratora sprzętowego:

- wykonanie specjalizowanego koprocesora realizującego ściśle określone zadanie lub jego najbardziej czasochłonne fragmenty; rozwiązanie kosztowne, wymagające wprawnego łączenia umiejętności inżynierskich i teoretycznych [3],
- projekt rekonfigurowalnego koprocesora [6], który zmienia swoją architekturę sprzętową stosownie do realizowanego zadania; zaletą tego rozwiązania jest możliwość wykorzystywania tej samej wersji koprocesora w różnych aplikacjach, przy czym właściwe rozłożenie algorytmu na podzadania pomiędzy sprzęt (koprocesor) i oprogramowanie (główny procesor systemu) pozwoli znacznie zwiększyć jego szybkość w porównaniu z koprocesorami uniwersalnymi o stałej architekturze.

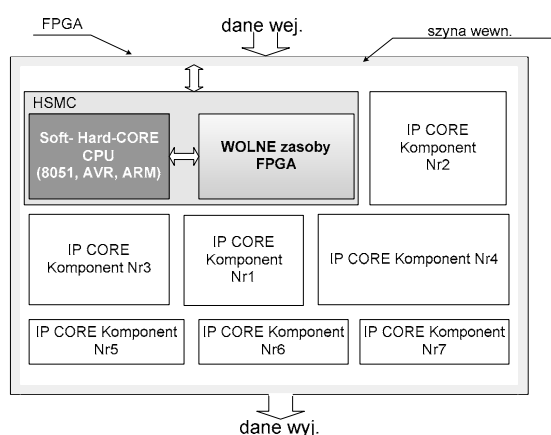
W praktyce inżynier w większości rozpatrywanych projektów, nie jest w stanie zagospodarować 100% logiki programowalnej układu FPGA w systemie zintegrowanym SOPC. Pozostaje pewien obszar niewykorzystanych zasobów sprzętowych (rysunek 2).



Rys. 2. Implementacja systemu w układzie SOPC

Proponuje się wykorzystać wolną logikę programowalną układu SOPC jako moduł wspomagający (koprocesor) pracę mikroprocesora, w celu zwiększenia wydajności głównej jednostki przetwarzania i sterowania. W ten sposób oznaczony zostaje mikrosystem cyfrowy – SPMC (Sprzętowo-Programowy Mikrosystem Cyfrowy), składający się z jednostki CPU i wolnej logiki programowalnej układu SOPC. W proponowanym rozwiązaniu mikrosystem cyfrowy pełni rolę głównej jednostki przetwarzania i sterowania w systemie. Część zadań wykonywanych do tej pory przez mikroprocesor, przejmowana jest przez część sprzętową.

Im więcej wolnych zasobów sprzętowych SOPC, tym więcej funkcji zostanie zaimplementowanych w dostępnej logice programowalnej – co korzystnie wpłynie na wydajność pracy całego systemu.



Rys. 3. Sprzętowo-programowy mikrosystem cyfrowy

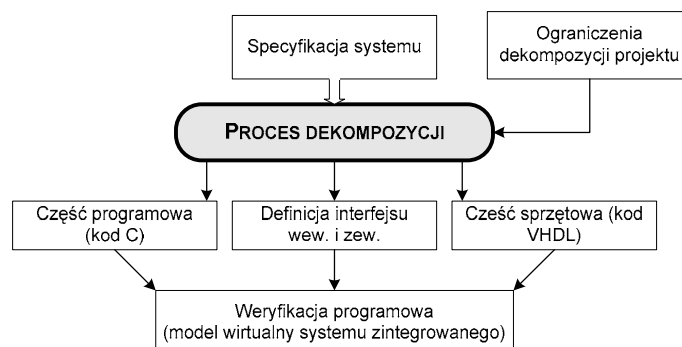
Utworzenie ściśle zintegrowanego, sprzętowo-programowego mikrosystemu cyfrowego, znacząco przyspiesza proces sterowania i przetwarzania danych [7].

Wzrost wydajności pracy systemu jest efektem:

- zrównoleglenia wykonywania wielorakich zadań wewnątrz implementacji sprzętowej,
- zrównoleglenia pracy jednostki sprzętowej z pracą jednostki programowej,
- skrócenia czasu przetwarzania danych i sterowania przez mniej obciążoną jednostkę programową.

3. DEKOMPOZYCJA SYSTEMOWA W POJĘCIU OGÓLNYM

Podstawowym etapem projektowania zintegrowanego jest proces dekompozycji systemowej [8,7]. Dekompozycja dotyczy podziału systemu pomiędzy sprzęt i oprogramowanie (rysunek 4). Konieczność wykonania dekompozycji determinuje system składający się z najmniej dwóch komponentów. Na podstawie specyfikacji wejściowej oraz zadanych wymagań (czasowych, użytkowych i innych) projekt dzielony jest na moduły realizowane przez mikroprocesor - część programowa (ang. *Software*) oraz układy programowalne współpracujące z procesorem - część sprzętowa (ang. *Hardware*).



Rys. 4. Dekompozycja funkcjonalna ADSO [10,11]

Podczas dekompozycji wykonywany jest optymalny podział na podstawie dedykowanego algorytmu podziału i przyjętych kryteriów. Wynikiem dekompozycji jest specyfikacja programowa oraz specyfikacja sprzętowa, które w połączeniu ze współbieżnymi mechanizmami komunikacji i sterowania [9] pozwalają na funkcjonalne i zarazem optymalne odwzorowanie pracy systemu w układach zintegrowanych. Istota procesu dekompozycji tkwi w spójności i integralności obu części wynikowych zarówno pod względem współpracy (wymiany informacji) poszczególnych elementów, jak i ich połączeń. Komponenty zakwalifikowane do jednej z list winny stanowić ściśle powiązany ze sobą podzbiór głównej sieci połączeń i elementów. Takie rozwiązanie minimalizuje czas potrzebny do wymiany informacji pomiędzy modułami sprzętowymi i programowymi, zwiększając w ten sposób częstotliwość pracy całego systemu.

4. METODOLOGIA PROJEKTOWANIA SPRZĘTOWO-PROGRAMOWEGO MIKROSYSTEMU CYFROWEGO

Metodologię projektowania systemów zintegrowanych, ze względu na poziom abstrakcji dekomponowanego systemu, można podzielić na dwie grupy:

- projektowanie systemowe, czyli dekompozycja systemowa; np. COSYMA, VulcanII,
- projektowanie procesorowe; np. procesory hybrydowe RISP, ASIP.

Projektowanie sprzętowo-programowego mikrosystemu cyfrowego ma charakter projektowania procesorowego. Metodologia projektowania procesorowego zorientowana jest na analizę i szacowanie składowych mikrosystemu, gdzie najmniejszymi blokami funkcyjnymi są pojedyncze instrukcje procesora (instrukcje programu, zadania). Algorytm

dekompozycji funkcjonalnej [7,10,11] dokonuje podziału zadań systemu przydzielając wyznaczone boki funkcyjne (instrukcje) do części programowej lub sprzętowej. Operacje analizy, szacowania, podziału i programowej współ-symulacji systemu realizowane są z użyciem modelu pośredniego, który determinuje zachowanie systemu, wszelkie aspekty jego pracy oraz właściwości funkcjonalne. Dla opracowanej architektury SPMC, jako model pośredni oznaczono sieci Petriego – zmodyfikowana podklasa sieci interpretowanych. Istotnym parametrem wybranego modelu pośredniego systemu jest gradient, czyli najmniejszy zbiór funkcji – kontener, poddawany procesowi analizy i szacowania. Definiując większe grupy funkcyjne możliwe jest pośrednie wspomaganie procesu automatycznej dekompozycji funkcjonalnej mikrosystemu cyfrowego. Projektant samodzielnie może scalać w bloki zadaniowe te procedury i funkcje, które pozostają w ścisłej kooperacji między sobą w systemie. Rozmiar gradientu – ilość funkcji wchodzących w skład „kontenera” – ma znaczący wpływ na proces analizy i szacowania w procesie dekompozycji funkcjonalnej. Mniejszy gradient – mniejsza precyzja analizy mikrosystemu cyfrowego, większy gradient – większa precyzja analizy. Precyzja analizy dotyczy przede wszystkim procesu implementacji układu FPGA. Ponadto podział specyfikacji mikrosystemu cyfrowego na część sprzętową (FPGA) i programową (program C) nie może zostać przeprowadzony bez wcześniejszych analiz systemu. Określenie rzeczywistych parametrów pracy poszczególnych kontenerów, zarówno w domenie programowej i sprzętowej ma znaczący wpływ na właściwą pracę dedykowanego algorytmu szacowania/podziału specyfikacji mikrosystemu cyfrowego. Analiza parametrów pracy bloków funkcyjnych musi zostać przeprowadzona dla zdefiniowanej architektury mikrosystemu cyfrowego. Dla części programowej - mikroprocesor (AVR, 8051, ARM, MicoBLAZE, inne), a dla części sprzętowej – rodzina i typ układu FPGA (Altera FLEX, Xilinx VIRTEX, inne).

LITERATURA

- [1] R. Hartenstein: *Reconfigurable Computing: the Roadmap to a new business model – and its impact on SoC design*, University of Kaiserslautern, Niemcy
- [4] [www.altera.com,http://www.altera.com/products/software/system/products/sopc/sop-index.html](http://www.altera.com/products/software/system/products/sopc/sop-index.html)
- [5] D.D. Gajski, F. Vahid: *Specification and Design of Embedded Hardware-Software Systems*, IEEE Design & Test of Computers, vol. 12, no 1, Spring 1995, pp. 53-67
- [2] A.A. Jerraya, J. Mermet: *System-Level Synthesis*, Kluwer Academic Publishers, 1999, Dordrecht/Boston/London

- [3] A. Österling, Th. Benner, R. Ernst, D. Herrmann, Th. Scholz, W.YeJ. Staunstrup, W. Wolf: *Hardware/Software Co-Design: Principles and Practice*, Kluwer Academic Publishers 1997.
- [6] F. Barat, R. Lauwereins, G. Deconinck *Reconfigurable Instruction Set Processors from a Hardware/Software Perspective*, IEEE Transactions on software engineering, vol. 28, no. 9, 2002
- [7] A. Stasiak: *System dekompozycji funkcjonalnej w projektowaniu zintegrowanym*, OWD'2003, Istebna, PPEE i BSE, ISBN 83-915991-5-9, Polska 2003
- [8] A. Stasiak: *System równoleglenia pracy mikrokontrolera sekwencyjnego dla sprzętowo-programowej architektury systemu osadzonego*, Materiały KKE'2003, wyd. Politechnika Koszalińska, Kołobrzeg 2003, Polska, str.375-380, ISBN 83-7365-030-X
- [9] A. Stasiak: *Metody synchronicznej wymiany danych w systemach osadzonych dla potrzeb dekompozycji systemowej*, W: Materiały XVIII Sympozjum Koła Zainteresowań Cybernetycznych. Warszawa, Polska, 2002 .- Warszawa : BEL Studio Sp. z o.o., 2002, s. 98--107 .- ISBN: 83-88442-39-2
- [10] A. Stasiak, M. Michalczak: *Eksperymentalny system CAD dla potrzeb zintegrowanego projektowania układów mikroprocesorowych*, Praca magisterska, Uniwersytet Zielonogórski, Wydział Elektrotechniki Informatyki i Telekomunikacji, Zielona Góra, lipiec 2002
- [11] A. Stasiak, M. Michalczak, Z. Skowroński: *Algorytm dekompozycji systemowej dla potrzeb projektowania zintegrowanego*, RUC 2003 : materiały VI krajowej konferencji naukowej. Szczecin, Polska, 2003 .- Szczecin : Politechnika Szczecińska, 2003, s. 113--122 .- ISBN: 83-87362-56-5